MENU SEARCH INDEX

E4584

1/1

JP-A-6-214843

JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11)Publication number: 06214843
(43)Date of publication of application: 05.08.1994

(51)Int.Cl.
G06F 12/00
G06F 12/00

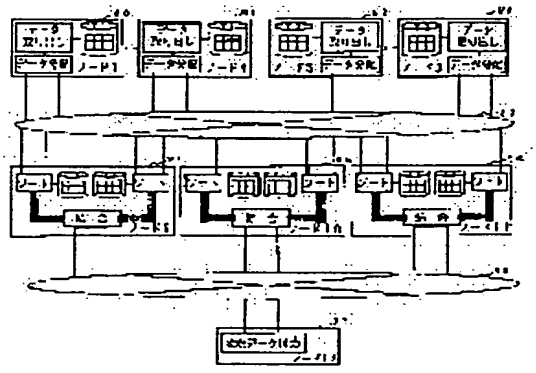
(21)Application number: 05007804
(22)Date of filing: 20.01.1993

(71)Applicant: HITACHI LTD
(72)Inventor: TSUCHIDA MASASHI
NAKANO YUKIO
KAWAMURA NOBUO
NEGISHI KAZUYOSHI
TORII SHUNICHI

(54) DATA BASE MANAGEMENT SYSTEM AND PROCESSING METHOD FOR INQUIRY

(57)Abstract:

PURPOSE: To provide an inquiry processing method and data base management system for accelerating inquiry processing.
CONSTITUTION: This data base management system, which is provided with plural nodes for data base processing and connects the plural nodes with the other nodes, is equipped with distributing nodes (nodes 1–8) provided with a storage means for distributedly storing the data base of an inquiry object and a distributing means for distributing information from the storage means to the other nodes, coupling nodes (nodes 9–11) provided with a rearranging means for distributed information, merge means for merging the information when there are plural kinds of information and collating means for inquiry based on the information, and decision managing node (node 12)



provided with an analyzing means for preparing the processing sequence of inquiry by receiving the inquiry and analyzing it, deciding means for deciding the distributing node and the coupling node to perform execution processing based on the analyzed result, and output means for outputting the result corresponding to the inquiry provided from the coupling node.

LEGAL STATUS

[Date of request for examination] 15.07.1996

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998 Japanese Patent Office

MENU SEARCH INDEX

E4584

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平6－214843

(43)公開日　平成６年(1994)８月５日

(51)Int.Cl.[5]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｇ０６Ｆ　12/00	５１３　Ｊ	8526－5B		
	５１２	8526－5B		

審査請求　未請求　請求項の数13　ＯＬ　（全 23 頁）

(21)出願番号　特願平5－7804

(22)出願日　平成５年(1993)１月20日

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地

(72)発明者　土田　正士
神奈川県川崎市麻生区王禅寺1099　株式会社日立製作所システム開発研究所内

(72)発明者　中野　幸生
神奈川県川崎市麻生区王禅寺1099　株式会社日立製作所システム開発研究所内

(72)発明者　河村　信男
神奈川県川崎市麻生区王禅寺1099　株式会社日立製作所システム開発研究所内

(74)代理人　弁理士　富田　和子

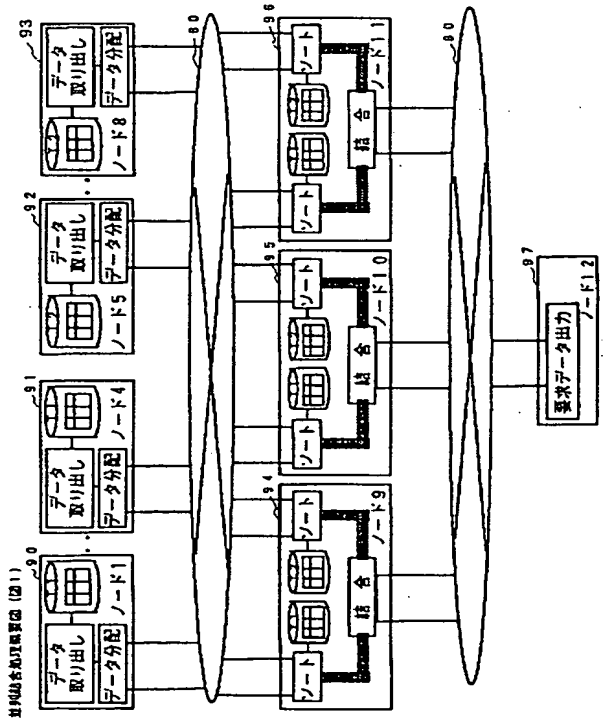
最終頁に続く

(54)【発明の名称】　データベース管理システムおよび問合せの処理方法

(57)【要約】　（修正有）

【目的】問合せ処理を高速化する問い合わせ処理方法およびデータベースシステムを提供する。

【構成】データベース処理をする複数のノードを備え、複数のノードは、他のノードと接続されるデータベース管理システムで、問い合わせ対象のデータベースを分散させて格納する記憶手段と、記憶手段からの情報を他のノードに分配する分配手段を備える分配ノード（ノード１～ノード８）と、分配された情報の並び替え手段と、その情報が複数ある場合にはそれらをマージするマージ手段と、その情報に基づいて問い合わせに対する突き合わせ手段とを備える結合ノード（ノード９～１１）と、問い合わせを受け付け、解析して問い合わせの処理手順を作成する解析手段と、解析結果に基づいて実行処理を行う分配ノードおよび結合ノードを決定する決定手段と、結合ノードから得られた、問い合わせに対する結果を出力する出力手段とを備える決定管理ノード（ノード１２）とを備える。



【特許請求の範囲】

【請求項１】データベース処理を実行する複数のノードを備え、該複数のノードは、ネットワークを介して他のノードと接続されるデータベース管理システムであって、

問い合わせ対象のデータベースを分散させて格納する記憶手段と、該記憶手段から情報を取り出して他のノードに取り出した情報を分配する分配手段を備える分配ノードと、

該分配ノードから分配された情報を並び替える並び替え手段と、該並び替えられた情報が複数ある場合にはそれらをマージするマージ手段と、該マージされた情報に基づいて問い合わせに対する突き合わせを実行する突き合わせ手段とを備える結合ノードと、

前記問い合わせを受け付けて、該問い合わせを解析して問い合わせの処理手順を作成する解析手段と、該解析手段の問い合わせの解析結果に基づいて実行処理を行う分配ノードおよび結合ノードを決定する決定手段と、前記結合ノードから得られた、問い合わせに対する結果を出力する出力手段とを備える決定管理ノードとを備えることを特徴とするデータベース管理システム。

【請求項２】請求項１において、前記決定手段は、前記解析手段の問い合わせの解析結果に基づいて前記分配ノードを決定し、前記分配ノードにおける予想される処理時間を算出し、該処理時間に基づいて結合ノードを決定することを特徴とするデータベース管理システム。

【請求項３】請求項２において、前記決定手段は、前記決定された分配ノードにおける予想される取り出し情報量に基づいて、前記結合ノードへの前記取り出し情報の分配を前記各結合ノードに均等に割当てるようにすることを特徴とするデータベース管理システム。

【請求項４】請求項３において、前記決定管理ノードは、前記決定手段において前記結合ノードに取り出し情報を均等に割当てるための、前記各ノードの記憶手段の情報に関する最適化情報を記憶している記憶手段を備えることを特徴とするデータベース管理システム。

【請求項５】請求項３において、前記決定管理ノードは、前記決定手段において前記結合ノードに取り出し情報を均等に割当てるためにあらかじめ定められたハッシュ関数を利用することを特徴とするデータベース管理システム。

【請求項６】請求項２において、前記複数のノードは、それぞれ独立に処理を行い、

前記結合ノードは、前記分配ノードからの分配された情報を逐次入力し、入力された情報ごとに処理を行うことを特徴とするデータベース管理システム。

【請求項７】請求項６において、前記分配ノードは、該分配ノードで分配する情報を並び替える並び替え手段をさらに有することを特徴とするデータベース管理システム。

【請求項８】請求項７において、前記決定手段は、前記分配ノードにおける予想される処理時間の算出結果から、より先に処理が終了する分配ノードに対して分配処理後に前記分配ノードの並び替え手段において並び替えをするように決定することを特徴とするデータベース管理システム。

【請求項９】請求項６において、前記決定手段は、前記処理時間に基づいて決定した前記結合ノードの台数を、所定数増加させるように決定することを特徴とするデータベース管理システム。

【請求項１０】請求項９において、前記結合ノードの並び替え手段は、並び替え処理の終了後にマージ処理をする機能を備えることを特徴とするデータベース管理システム。

【請求項１１】請求項６において、前記突き合わせ手段は、前記マージ処理をする機能を備えることを特徴とするデータベース管理システム。

【請求項１２】請求項１１において、前記決定手段は、前記突き合わせ手段および前記出力手段における予想される処理時間を算出し、該算出結果に基づいて、前記出力手段の処理時間が、前記突き合わせ手段の処理時間より大きい場合には、前記突き合わせ手段に前記マージ処理を行わせるように決定することを特徴とするデータベース管理システム。

【請求項１３】情報を記憶する記憶手段を備え、前記記憶手段から情報を取り出して他のノードに取り出した情報を分配し、並び替えを実行する分配ノード群と、

該分配ノードから分配された情報を並び替え、該並び替えられた情報が複数ある場合にはそれらをマージし、該マージされた情報に基づいて問い合わせに対するマージと突き合わせとをする結合ノード群と、

前記問い合わせを受け付けて、該問い合わせを解析して問い合わせの処理手順を作成し、該解析手段の問い合わせの解析結果に基づいて、実行処理を行う分配ノードおよび結合ノードを決定する決定管理ノード群とを備えるデータベース管理システムにおける問い合わせ処理方法であって、

前記決定管理ノードは、前記解析手段の問い合わせの解析結果に基づいて前記分配ノードを決定し、前記分配ノードにおける予想される処理時間を算出し、該処理時間に基づいて結合ノードを決定し、

前記決定された分配ノードのそれぞれは、前記問い合わせの解析結果に基づいて前記記憶手段から情報を取り出して該取り出した情報を他のノードに分配し、並び替えを行い、

前記決定された結合ノードのそれぞれは、前記分配ノードから分配された情報を並び替え、該並び替えられた情報が複数ある場合にはそれらをマージし、該マージされた情報に基づいて問い合わせに対するマージと突き合わせとをし、

前記結合ノードから得られた、問い合わせに対する結果を出力することを特徴とする問合せの処理方法。

【発明の詳細な説明】

【０００１】

【産業上の利用分野】本発明は、データベース処理装置に関し、特に、リレーショナルデータベース管理システムに適した問合せの並列処理に好適な問合せ処理方法に関する。

【０００２】

【従来の技術】データベース管理システム（以下ＤＢＭＳと略記）、特に、リレーショナルＤＢＭＳは、非手続き的な言語で表現された問合せを処理し、内部処理手順を決定し、内部処理手順に従って実行する。このデータベース言語としては、ＳＱＬが用いられる（Database Language SQL ISO 9075:1989)。従来の問合せ処理の主な方法には、予め設定した規則に基づいて単一の内部処理手順を決定するものと、各種統計情報を用いて選定された複数の候補処理手順から、コスト評価により、最適と思われるものを決定するものとがある。前者は、処理手順作成のための負荷は小さいけれども、一律に設定された規則の妥当性に問題があり、選ばれた内部処理手順の最適性にも問題がある。後者は、各種統計情報の管理し、複数の候補処理手順の作成し、それらのコスト評価のための負荷を算出して最適な処理手順を与える。上記両者の組合せ技術としては、例えば、Satoh, K., et. al." Localand Global Optimization Mechanisms for Relational Database", Proc. VLDB, 1985. がある。該従来技術では、問い合わせの条件からデータ量を推定して処理手順を決めている。

【０００３】また、多くのＤＢＭＳは、問合せ解析処理と問合せ実行処理との２フェーズの処理を経て、問合せ処理が実現される。ホスト言語（ＣＯＢＯＬ、ＰＬ／Ｉ等）に問合せ言語を組み込む場合、当アプリケーションプログラム実行前に予め問合せを問合せ解析処理し、実行形式である１つの内部処理手順を作成している。この問合せ表現では、多くの場合、検索条件式にはホスト言語の変数が記述される。この変数に定数が代入されるのは、既に問合せ解析処理された結果の内部処理手順の実行時、すなわち、問合せ実行時である。この場合の問題点としては、変数に代入される値に従って複数の最適な処理手順が考えられることである。この問題を解決するために、問合せ実行処理時に複数の処理手順を作成しておき、問合せ実行時に変数に代入された値に従って処理手順を選択するものがある。コードの技術に関するものとしては、特開平１－１９４０２８号公報、および、Graefe, G., et. al."Dynamic Query Evaluation Plans", Proc. ACM-SIGMOD, 1989. に記載されている技術がある。

【０００４】さらに、ＣＰＵ性能、ディスク容量の延びを上回るような、トランザクション量の増大、データベース量の増大に対応して、スケーラブルな並列データベースシステムの提供がユーザから望まれている。データベースシステムに対するユーザの性能要件として、数万を超える同時実行ユーザ数への対応、テラバイト単位の検索トランザクションの出現、表サイズに比例しない応答時間の保証がある。並列データベースシステムは、近年のハードウェアコストの低減と相まって、注目を浴びている。並列データベースシステムについては、DeWitt, D., et. al. :"Parallel Database Systems: The Future of High Performance Database Systems", CACM, Vol. 35, No. 6, 1992. に記載の技術がある。そのようなシステムでは、密結合あるいは疎結合にプロセッサを接続し、データベース処理を複数のプロセッサに静的／動的に処理を配分し、スケジュールする必要がある。並列度を増せば応答性能は向上するが、過度の並列度は逆にオーバヘッドの増大、他トランザクションの応答時間の延び等の影響がある。そのため、適度な並列度の設定が重要である。

【０００５】データベース処理において、処理対象となるデータは、二次記憶装置上に存在し、各データベース演算に対して大量データの読み出しおよび転送が必要となる。並列データベースシステムにおいても、転送するデータが大量となる場合、データ転送時間がデータベースシステムの性能ネックとなる。そこで、二次記憶装置からデータを転送する時間を有効活用する方法が考えられる。これは、データの転送時間と当該データに対するデータベース処理に要する時間とをオーバラップさせるものであり、従来技術として良く知られている。この方式は、相互結合ネットワークで接続されるプロセッサ群間のデータ転送にも適用可能である。

【０００６】

【発明が解決しようとする課題】上記従来技術において、問合せ最適化処理とは、ユーザが入力した問合せからデータベースシステムの各種統計情報を基にし、最も効率の良い処理手順をＤＢＭＳが自動判定するものである。さらに、問合せの選択条件式に変数が埋め込まれている場合には、複数の処理手順を問合せ解析時に展開しておき、問合せ実行時に当変数に代入される値に従って処理手順を選択することによって、最適な処理手順が選択される。

【０００７】並列データベース処理では、各ノード（プロセッサあるいはプロセッサとディスク装置との対）へデータベース演算が分割され、各ノードで各データベース演算が並列にあるいはパイプライン的に動作する。上記従来技術によれば、この並列処理形態でも、各ノードで処理手順を選択する方法は適用可能である。

【０００８】しかし、並列に動作する処理では、同時間にそれぞれのノードが並行処理をするが、各ノードで実行するデータベース演算に対応して各ノード数を決定できないという問題がある。すなわち、ノード数を決定する基準が明確でないために、過度の並列化は逆にオーバ

ヘッドの増大等の影響があり、最適に負荷分散することが困難である。

【０００９】また、パイプライン動作させる処理では各ノードへデータベース演算が分割格納されるが、データの分割にバラツキが存在する場合、各ノードへの均等分割方法が明確でない。

【００１０】さらに、処理時間の制約があったときなどのように、その時間内で複数の処理を行う場合において、各ノードで実行する各データベース演算をパラメータ化し、期待する処理時間に基づいて時間調整（チューニング）をする方法も明確でない。

【００１１】本発明の目的は、問合せ処理を高速化する問い合わせ処理方法およびデータベースシステムを提供することにある。

【００１２】

【課題を解決するための手段】本発明は、上記課題を解決するために、データベース処理を実行する複数のノードを備え、該複数のノードは、ネットワークを介して他のノードと接続されるデータベース管理システムであって、問い合わせ対象のデータベースを分散させて格納する記憶手段と、該記憶手段から情報を取り出して他のノードに取り出した情報を分配する分配手段を備える分配ノードと、該分配ノードから分配された情報を並び替える並び替え手段と、該並び替えられた情報が複数ある場合にはそれらをマージするマージ手段と、該マージされた情報に基づいて問い合わせに対する突き合わせを実行する突き合わせ手段とを備える結合ノードと、前記問い合わせを受け付けて、該問い合わせを解析して問い合わせの処理手順を作成する解析手段と、該解析手段の問い合わせの解析結果に基づいて実行処理を行う分配ノードおよび結合ノードを決定する決定手段と、前記結合ノードから得られた、問い合わせに対する結果を出力する出力手段とを備える決定管理ノードとを備える。

【００１３】前記決定手段は、前記解析手段の問い合わせの解析結果に基づいて前記分配ノードを決定し、前記分配ノードにおける予想される処理時間を算出し、該処理時間に基づいて結合ノードを決定することができる。

【００１４】前記決定手段は、前記決定された分配ノードにおける予想される取り出し情報量に基づいて、前記結合ノードへの前記取り出し情報の分配を前記各結合ノードに均等に割当てるようにする。

【００１５】前記決定管理ノードは、前記決定手段において前記結合ノードに取り出し情報を均等に割当てるための、前記各ノードの記憶手段の情報に関する最適化情報を記憶している記憶手段を備えることができる。

【００１６】前記決定管理ノードは、前記決定手段において前記結合ノードに取り出し情報を均等に割当てるためにあらかじめ定められたハッシュ関数を利用する。

【００１７】また、前記複数のノードは、それぞれ独立に処理を行い、前記結合ノードは、前記分配ノードからの分配された情報を逐次入力し、入力された情報ごとに処理を行う。

【００１８】さらに、前記分配ノードは、該分配ノードで分配する情報を並び替える並び替え手段を有するようにしてもよい。

【００１９】前記決定手段は、前記分配ノードにおける予想される処理時間の算出結果から、より先に処理が終了する分配ノードに対して分配処理後に前記分配ノードの並び替え手段において並び替えをするように決定することができる。

【００２０】前記決定手段は、前記処理時間に基づいて決定した前記結合ノードの台数を、所定数増加させるように決定する。

【００２１】前記結合ノードの並び替え手段は、並び替え処理の終了後にマージ処理をする機能を備えるようにしてもよい。前記突き合わせ手段は、前記マージ処理をする機能を備えるようにしてもよい。

【００２２】前記決定手段は、前記突き合わせ手段および前記出力手段における予想される処理時間を算出し、該算出結果に基づいて、前記出力手段の処理時間が、前記突き合わせ手段の処理時間より大きい場合には、前記突き合わせ手段に前記マージ処理を行わせるように決定する。

【００２３】

【作用】前記決定管理ノードは、前記解析手段の問い合わせの解析結果に基づいて前記分配ノードを決定し、前記分配ノードにおける予想される処理時間を算出し、該処理時間に基づいて結合ノードを決定する。決定手段は、前記決定された分配ノードにおける予想される取り出し情報量に基づいて、前記結合ノードへの前記取り出し情報の分配を前記各結合ノードに均等に割当てるようにする。

【００２４】前記決定された分配ノードのそれぞれは、前記問い合わせの解析結果に基づいて前記記憶手段から情報を取り出して該取り出した情報を他のノードに分配する。分配ノードおよび結合ノードは、それぞれ独立に処理を行い、前記結合ノードは、前記分配ノードからの分配された情報を逐次入力し、入力された情報ごとに処理を行う。前記決定された結合ノードのそれぞれは、前記分配ノードから分配された情報を並び替え、該並び替えられた情報が複数ある場合にはそれらをマージし、該マージされた情報に基づいて問い合わせに対する突き合わせをし、前記結合ノードから得られた、問い合わせに対する結果を出力する。

【００２５】また、決定手段は、前記分配ノードにおける予想される処理時間の算出結果から、より先に処理が終了する分配ノードに対して分配処理後に前記分配ノードの並び替え手段において並び替えをするように決定する。決定された分配ノードの並び替え手段は、該分配ノードで分配する情報を並び替える。

【００２６】さらに、決定手段は、前記突き合わせ手段および前記出力手段における予想される処理時間を算出し、該算出結果に基づいて、前記出力手段の処理時間が、前記突き合わせ手段の処理時間より大きい場合には、前記突き合わせ手段に前記マージ処理を行わせるように決定する。決定された突き合わせ手段は、マージ処理をする。

【００２７】また、処理時間があらかじめ定まっているときに、該処理時間以内で処理をさせるために、前記決定手段は、分配ノードにおける予想される処理時間に基づいて決定した前記結合ノードの台数を、所定数増加させるように決定する。これにより、結合ノードの台数が増加し、結合ノードの並び替え手段は、並び替え処理が短時間で処理できるので、並び替え処理の終了後にマージ処理をする。

【００２８】本発明の問合せ処理方法によれば、各ノードで実行するデータベース演算に対応して各ノード数を決定できる。また、データの分割にバラツキが存在する場合、各ノードへデータを均等に分割させ、各ノードで実行する各データベース演算をパラメタ化し期待する処理時間均等化させるので、各ノード間で処理時間の偏りがなく、円滑にパイプライン動作させることが可能である。

【００２９】

【実施例】以下、本発明の実施例を図面に基づいて詳細に説明する。

【００３０】図２は、本実施例のデータベースシステムの概念図を示している。図２において、データベースシステムは、ユーザが作成した、複数のアプリケーションプログラム（以下、ＡＰと略記する）１０および１１と、問合せ処理やリソース管理等データベースシステム全体の管理を行うＤＢＭＳ２０と、データベース処理において、入出力処理対象となるデータの読書きを行い、計算機システム全体の管理を受け持つオペレーティングシステム（以下では、オペレーティングシステムをＯＳと略記する）３０と、データベース処理対象となるデータを格納するデータベース４０と、データベースの定義情報を管理するディクショナリ５０とを有する。ＤＢＭＳ２０は、他のデータベース管理システムと接続されている。ディクショナリ５０には、本実施例において使用する結合カラムに関する最適化情報なども記憶されている。

【００３１】上記ＤＢＭＳ２０は、システム全体の管理、制御に加えて、入出力の管理等を行うシステム制御部２１と、問い合わせに関する論理処理を行う論理処理部２２と、データベースの物理処理を実行する物理処理部２３と、当ＤＢＭＳ２０で処理対象となるデータを格納するデータベースバッファ２４とを備える。また、論理処理部２２は、問合せの構文解析、意味解析を行う問合せ解析２２０、適切な処理手順を生成する静的最適化処理２２１、処理手順に対応したコードの生成を行なうコード生成２２２、静的最適化処理２２１で生成された処理手順候補から最適なものを選択する動的最適化処理２２３、および、当コードの解釈実行を行うコード解釈実行部２２４を備える。また、物理処理部２３は、アクセスしたデータの条件判定、編集、レコード追加等を実現するデータアクセス処理２３０、データベースレコードの読み書きを制御するデータベースバッファ制御２３１、入出力対象となるデータの格納位置を管理するマッピング処理２３２、および、システムで共用するリソースの排他制御を実現する排他制御２３３を備える。

【００３２】図３は、本発明が適用されるハードウェア構成の一例を示すものである。具体的には、図３は、プロセッサおよびディスク装置が１ノードを構成し、複数のノードを備える並列プロセッサシステムの適用構成例を示している。図３において、プロセッサ６０～６５およびディスク装置７０～７５が相互結合ネットワーク８０で接続される。図３に示すハードウェア構成は、図２に示すデータベースシステムを複数のプロセッサで並列処理するための構成であり、各ノードに対してそれぞれ処理が分散される。

【００３３】上記各ノードごとに機能分散した場合の構成を図１に示す。図１は、本実施例が適用されたデータベースシステムの概要図を示している。以下に、並列データベースシステムの処理例を図１を参照して説明する。この例では、データベースに対する検索要求に並列処理を適用する。図１において、各ノードは、データを取り出して分配処理するソート機能と、複数のノードでそれぞれソートされたデータを結合処理するマージ機能とが各ノードごとに割り当てられている。ノードによりソート機能だけを備えるものや、ソート機能とマージ機能とを備えるものがある。データベースは、ユーザから２次元のテーブル形式で見られる表から成るものとし、当該表は行あるいはロウごとにデータが存在するものである。また、ロウは、１つ以上の属性（これを「カラム」という）からなる。図１においては、データベースの表としてＴ１およびＴ２があり、ノード１（９０）からノード４（９１）に表Ｔ１が、ノード５（９２）からノード８（９３）に表Ｔ２が各々格納されており、これらの各ノードが分配ノードであり、分配ノードにおいて格納している表に基づいてデータ取り出し処理およびデータ分配処理が実行される。また、ノード９（９４）からノード１１（９６）は、結合ノードであり、ノード１～４およびノード５～８から出力されるデータを受け取り、部分列ソート処理およびマージ処理をして完全列の作成を実行する。さらに、ノード１２（９７）は、問い合わせを受け付け、該問い合わせを解析し、問い合わせに対する処理を実行する分配ノードおよび結合ノードの数を決定する決定ノードである。また、ノード１２（９７）は、ノード９～１１から出力されたデータを受け取

り出力する。これらのノード群は、相互結合ネットワーク８０で接続され、ノード１～４およびノード５～８と、ノード９～１１とが並列に動作し、しかもノード１～４およびノード５～８でそれぞれ処理された結果は、すぐにノード９～１１で処理を行うというようなパイプライン的に動作する（以下、並列パイプライン動作と呼ぶ）。また、ノード９～１１とノード１２とも同様にパイプライン動作する。以下では、ノード９～１１における部分列ソート処理をスロットソート処理といい、完全列作成処理をNウェイマージ処理と呼ぶ。スロットソート処理は、データが格納されるページを対象とするページ内のソート処理を指し、スロット順に読みだせば昇順にロウがアクセス可能とする。Nウェイマージ処理は、Nウェイのバッファを用いて、各マージ段でN本のソート連を入力にして最終的に１本のソート連を作成する。

【００３４】データベース検索処理のための問合せは、例えば、以下のようになる。

【００３５】SELECT　T1．C3，T2．C3
FROM　　T1，T2
WHERE　T1．C1＝T2．C1
AND　　T1．C2＝？

このような問い合わせが、ノード１２において受け付けられると、ノード１２において、最適な分配処理方法が選択され、各ノードに対してネットワークを介して指示される。上記の問い合わせにおいては、ノード１（９０）からノード４（９１）に表Ｔ１が、ノード５（９２）からノード８（９３）に表Ｔ２が各々格納されているので、各ノードにおいてデータ取り出し処理およびデータ分配処理が実行される。また、ノード９（９４）からノード１１（９６）では、ノード１～４およびノード５～８から出力されるデータを逐次受け取り、ソート処理および結合処理を実行する。ノード１２（９７）では、ノード９～１１から出力されたデータを受け取り出力する。これによりデータベース検索は終了する。

【００３６】つぎに、上記各ノードの処理時間の関係について図４を参照して説明する。図４は、並列パイプライン動作を説明するための概要図を示す。図４において、１００および１０１は、図１におけるノード１（９０）からノード４（９１）と、ノード５（９２）からノード８（９３）とにおける処理に対応し、データ取り出し処理およびデータ配分処理を実行する。１１０および１１１は、ノード９（９４）からノード１１（９６）における処理に対応し、スロットソート処理、Nウェイマージ処理、突き合わせ処理が実行される。１２０は、ノード１２（９７）における処理に対応し、要求データ出力処理が実行される。時間軸に沿えば、データ取り出し処理およびデータ配分処理１００および１０１で処理されたデータは、逐次スロットソート処理１１０および１１１に移り、パイプライン的に実行される。データ取り出し処理からスロットソート処理までを取り出しフェーズと呼ぶ。また、Nウェイマージ処理１１０および１１１は、それぞれのノードで単に並列に実行される。このNウェイマージ処理期間をマージフェーズと呼ぶ。さらに、突き合わせ処理の結果は要求データ出力処理１２０に逐次転送されてパイプライン的に実行される。この突き合わせから要求データ出力までを結合フェーズと呼ぶ。図４に示すタイムチャートは、図１に示す問合せ例を適用した場合の処理内容である。取り出しフェーズにおいて、ノード１（９０）からノード４（９１）における処理時間は、Ｔ１データ取り出し／データ分配処理時間１３０として示す。また、ノード５（９２）からノード８（９３）における処理時間は、Ｔ２データ取り出し／データ分配処理時間１３１で示し、相互結合ネットワーク８０における転送時間をデータ分配転送時間１４０で示し、ノード９（９４）からノード１１（９６）における処理時間は、Ｔ１／Ｔ２スロットソート処理時間１５０に示すように、各々実行される。図４において、取り出しフェーズは、スロットソート処理完了待ち合わせ１８０の時点までで終了する。また、マージフェーズは、ノード９（９４）からノード１１（９６）における処理時間は、Ｔ１／Ｔ２Nウェイマージ処理時間１５１に示す時間において実行される。このマージフェーズは、Ｔ１／Ｔ２Nウェイマージ処理待ち合わせ１８１までで終了する。結合フェーズは、ノード９（９４）からノード１１（９６）における処理時間は、突き合わせ処理時間１５２で示し、相互結合ネットワーク８０における処理時間は、結合結果転送時間１６０で示し、ノード１２（９７）における処理時間は、要求データ出力処理時間１７０で示し、各々その時間内に実行される。

【００３７】つぎに、図１におけるノード１２の各ノード群への処理の振り分け方法について図５を参照して説明する。図５は、データ分配処理における各ノード群への振り分け方法を示す説明図である。前提として、データ取り出し／データ分配処理をするノード群は、プロセッサ２００～２３０とディスク装置２０１～２３１とを備えるノード１～１０の１０台からなる。また、結合処理をするノード群は、プロセッサ２４０～２５０とディスク装置２４１～２５１とを備えるノード１１～１５の５台からなるとする。ディクショナリ５０には、結合カラムに関する最適化情報５１が格納されていている。該最適化情報５１とは、データベースのデータを均等に分割するための情報であり、例えば、結合カラムに対するデータ件数は通常均一でないので、データ件数が均一になるように結合カラムで分割するようにするものである。図５に示すように、ノード１～１０に格納されているデータが、ｖ１からｖ１０の各分割範囲で均等にデータ分割可能であることを示す。この場合、ノード１１～１５に均等にデータ分割するためにはｖ１～ｖ２、ｖ３～ｖ４、ｖ５～ｖ６、ｖ７～ｖ８、ｖ９～ｖ１０の５区間にそれぞれノード番号１１、１２、１３、１４、１５

を対応付けるような分配処理手段を備えればよい。上記最適化情報が存在しない場合、適当なハッシュ関数を設定してデータ分配を行なえばよい。このようにして、図１におけるノード１２では、分配処理手段を備えることにより、Ｎウェイマージ処理を行う際の各ノード群への処理の振り分けを行う。これにより、上記のような場合には、ノード１１～１５に均等にデータ分割することができ、処理時間が均等になる。

【００３８】つぎに、Ｎウェイマージ処理を行う際の結合ノード数の決定方法について図６を参照して説明する。図６は、結合ノード数決定方法を説明するための概要図を示している。

【００３９】図１における並列結合処理の各フェーズ、各処理の処理時間をグラフ化し、図４に示す並列パイプライン動作概要に合わせてレイアウトしている。図６において、データ取り出し／データ分配処理が、ノード１～８で実行され、３００～３０５の処理時間がそれぞれかかるものとする。ここでは、ノード５の処理時間３０４が最大処理時間であるとする。スロットソート処理時間は、結合処理ノード数Ｎと、予め決められたシステム特性（ＣＰＵ性能、ディスク装置性能等）と、データベース演算方法とから導けることができ、スロットソート処理の性能特性は一般的に下記に示すような式で求めることができる。

【００４０】

【数１】Ｅ＝ａ／Ｎ＋ｂ＊Ｎ＋ｃ

パイプライン処理を行う際の効果を最大にするために、スロットソート処理の性能特性と最大処理時間３０４との交点となる結合ノード割当て数３５０をノード数として求めることができる。結合ノード割当て数３５０が決まると、Ｎウェイマージ処理時間３２０および突き合わせ処理時間３３０が、Ｎウェイマージ処理の性能特性と突き合わせ処理の性能特性とから同様に推定できる。これらの処理時間の合計が問い合わせに対する全体の処理時間となる。このように結合ノード数を決定し、データ取り出し／データ分配処理において分配されたデータを逐次マージして同時に処理することにより、全体の処理時間（問い合わせをしてから出力されるまでの応答時間）を短縮することができる。

【００４１】結合ノード数を決定する場合に用いられる性能特性の具体例を以下に示しておく。例えば、ロウ数が表Ｔ１および表Ｔ２とも１０、０００、０００件あり、条件数がＴ１－１コ（全体ロウが１％に絞られる）とし、データ取り出し／データ分配処理をする分配ノード数が表Ｔ１および表Ｔ２ともそれぞれ１６ノードで均等分割され、結合ノード数が８ノードで、プロセッサ性能が５０ＭＩＰＳ（１秒間に５千万命令実行）で、ネットワーク転送レートが２０Ｍバイト／秒であるとする。このような条件で実際のデータベース管理システムに処理させた結果もしくは性能モデルから算出した結果が以下のようになる。

【００４２】表Ｔ１および表Ｔ２の分配ノードの処理時間がそれぞれ１８０秒、Ｔ１／Ｔ２スロットソート処理時間が８０秒、Ｎウェイマージ処理時間が３８０秒、突き合わせ処理時間が１１０秒、要求データ出力時間が１０秒となる。これらの結果の処理性能に基づいて問い合わせに対する処理時間を推定する。

【００４３】つぎに、図６に示した結合ノード数決定方法を基にして、応答時間をさらに短縮するための処理時間調整方法（チューニング方法）について、図７、図８および図９を参照して説明する。以下に示す方法は、上記ノード１２の分配処理手段において、各ノード群への処理の振り分けを決定する際にあらかじめ算出されて、その結果より振り分けを決定するものである。

【００４４】図７は、スロットソート前処理化の概要図を示す。データ取り出し／データ分配処理が、ノード１～８で実行され、各３００～３０５の処理時間がそれぞれかかるものとする。ノードごとの処理時間には各表のデータ数によりバラツキが存在する。また、スロットソート処理は、結合処理ノード群で実行されるように設定されている。ノードごとの処理時間でバラツキがある場合には、データ取り出し／データ分配処理ノード群へスロットソート処理を移す処理手順を考える。図７に、スロットソートの前処理化として示すように、データ取り出し／データ分配処理がより早く終了したノードでスロットソート処理を行う。その処理によれば、結合ノード割当て数３５０のノードにおけるスロットソート処理時間が３１０から３１２に削減できる。その処理時間の差３１１においてＮウェイマージ処理を移す。これは、スロットソート処理の連長を延ばすことにほかならない。これによって、Ｎウェイマージ処理時間が削減でき、結果的に応答時間が削減できる。

【００４５】図８は、スロットソート連長チューニング概要図を示している。例えば処理時間の制約があったときなどのように、その時間内で複数の処理を行う場合において、各ノードで実行する各データベース演算をパラメータ化し、期待する処理時間に基づいて時間調整（チューニング）をする方法について説明する。図６で求まる結合ノード割当て数３５０から最小限だけ結合処理ノードを増やし、応答時間の短縮を図る。この場合の結合ノード割当て数を３５１とする。結合ノード割当て数３５１とすると、スロットソート処理時間は３１０から３１２へ削減される。パイプライン効果を最大にするため、処理時間３１１においてＮウェイマージ処理をスロットソート処理へ移す。これによって、Ｎウェイマージ処理のマージ回数が減り、処理時間が３２０と削減でき、結果的に応答時間が削減できる。

【００４６】図９は、Ｎウェイマージ回数チューニングの概要図を示す。結合ノード割当て数３５０で決まる突き合わせ処理時間３３０が要求データ出力処理時間３４

0より小である場合には、Nウェイマージ処理の最終段のマージ処理を突き合わせ処理に移すようにできる。Nウェイマージ処理の最終段のマージ処理時間３３１と突き合わせ処理時間３３０との和が要求データ出力処理時間３４０を上回らなければ、当最終段のマージ処理を突き合わせ処理へ移す。これによって、応答時間が削減できる。

【００４７】つぎに、本実施例におけるデータベース管理システムの動作フローを説明する。図１０、図１１、図１２、図１３、図１４および図１５は、本実施例におけるＤＢＭＳの処理のフローチャートを示す。図１０において、ＤＢＭＳは、問合せ実行前に行われる問合せの解析処理（ステップ２２０）、静的最適化処理（ステップ２２１）およびコード生成（ステップ２２２）により問い合わせ解析を行う問合せ解析処理４００と、変数に定数を代入し、処理手順を選択する動的最適化処理（ステップ２２３）および問合せのコード解釈実行（ステップ２２４）により問い合わせに対する実行処理を行う問合せ実行処理４１０とを行う。

【００４８】以下、各処理部の概要について述べる。

【００４９】（ａ）問合せ解析処理４００

図１０（ａ）および（ｃ）において、問合せ解析（ステップ２２０）では、上記ノード１２においてアプリケーションプログラムにより入力された問合せ文の構文解析、意味解析を実行する（ステップ２２００）。図１０（ａ）において静的最適化処理（ステップ２２１）では、上記ノード１２において問合せで出現する条件式から条件を満足するデータの割合を推定し、予め設定している規則を基に、有効なアクセスパス候補（特にインデクスを選出する）を作成し、処理手順の候補を作成する。コード生成（ステップ２２２）では、上記ノード１２において処理手順候補を実行形式に展開する。

【００５０】（ｂ）問合せ実行処理４１０

図１０（ｂ）において、動的実行時最適化（ステップ２２３）では、上記ノード１２において代入された定数に基づき、各ノード群で実行する処理手順を決定する。コード解釈実行（ステップ２２４）では、それぞれのノードにおいて処理手順を解釈し、実行する。

【００５１】つぎに、各処理部の詳細な処理フローの説明を行う。

【００５２】図１０（ｄ）において、動的最適化処理（ステップ２２１）では、問合せに出現する条件式の述語選択率推定し（ステップ２２１０）、インデクス等からなるアクセスパスの剪定をし（ステップ２２１１）、これらアクセスパスを組合せた処理手順候補の生成をする（ステップ２２１２）。

【００５３】図１０（ｅ）において、述語選択率推定（ステップ２２１０）では、問合せ条件式に変数が出現するか否かチェックする（ステップ２２１０１）。変数が出現すれば、当条件式にカラム値分布情報があるかチェックする（ステップ２２１０４）。存在すれば終了する。存在しなければ、条件式の種別に応じてディフォルト値を設定し（ステップ２２１０５）、終了する。変数が出現しなければ、当条件式にカラム値分布情報があるかチェックする（ステップ２２１０４）。存在しなければ、条件式の種別に応じてディフォルト値を設定し（ステップ２２１０５）、終了する。存在すれば、カラム値分布情報を用いて選択率を算出する（ステップ２２１０３）。

【００５４】図１１において、アクセスパス剪定２２１２では、問合せ条件式で出現するカラムのインデクスをアクセスパス候補として登録する（ステップ２２１２０）。つぎに、問合せでアクセス対象となる表が複数ノードに分割格納されているかチェックする（ステップ２２１２１）。分割格納されていれば、パラレルテーブルスキャンをアクセスパス候補として登録する（ステップ２２１２３）。分割格納されていなければ、テーブルスキャンをアクセスパス候補として登録する（ステップ２２１２３）。各条件式の選択率が既に設定済みか否かチェックする（ステップ２２１２４）。設定済みであれば、各表に関して選択率が最小となる条件式のインデクスをアクセスパスの最優先度とする（ステップ２２１２５）。設定済みでなければ、各条件式の選択率の最大値／最小値を取得する（ステップ２２１２６）。最後に、ＣＰＵ性能、ＩＯ性能等のシステム特性より各アクセスパスの選択基準を算出し（ステップ２２１２７）、単一あるいは複数のインデクスを組合せたアクセスパスでの選択率が上記選択基準を下回るものだけアクセスパス候補として登録する（ステップ２２１２８）。

【００５５】図１２において、処理手順候補生成２２１３は、問合せでアクセス対象となる表が複数ノードに分割格納されているかチェックする（ステップ２２１３０）。分割格納されていれば、ステップ２２１３５へ移行する。分割格納されていなければ、処理手順候補にソート処理が含まれているか否かをチェックする（ステップ２２１３１）。含まれていれば、ステップ２２１３５へ移行する。含まれていなければ、問合せでアクセス対象となる表のアクセスパスが唯一であるかチェックし（ステップ２２１３２）、唯一であれば単一の処理手順を作成し（ステップ２２１３３）、唯一でなければ複数の処理手順を作成し（ステップ２２１３４）、終了する。ステップ２２１３５では、結合可能な２ウェイ結合へ問合せを分解する。分割格納される表の格納ノード群に対応して、データ読みだし／データ分配処理手順を候補として登録する。また、スロットソート処理手順を候補として登録する（ステップ２２１３６）。結合処理ノード群に対応して、スロットソート処理手順、Ｎウェイマージ処理手順および突き合わせ処理手順を候補として登録し、スロットソート連長およびマージ処理回数をパラメタ化しておく（ステップ２２１３７）。要求データ